



TITLE:

# String Controlled Grammar (情報科学の数学的理論)

AUTHOR(S):

伊藤, 英則; 稲垣, 康善; 福村, 晃夫

---

CITATION:

伊藤, 英則 ...[et al]. String Controlled Grammar (情報科学の数学的理論).  
数理解析研究所講究録 1971, 123: 103-113

ISSUE DATE:

1971-09

URL:

<http://hdl.handle.net/2433/106502>

RIGHT:

## String Controlled Grammar

名古屋大学 工学部

伊 藤 英 則

稻 垣 康 善

福 村 晃 夫

### § 1. はじめに

生成規則の適用順序を制御する文法が最近いろいろ考えられている。これらは異なった制御の方法を用いているが、いずれも複雑さの低いクラス of 文法に制御を加えることにより高いクラス of 言語を生成できる特徴を持っている。

そこで一つの新しい試みとして、CF 文法の文法規則の適用順序を補助記号の系列によって制御する *String Grammar* を考えた。これは *Core* 部と *String* 部を有していて、*Core* 部には CF 文法の生成規則を持ち、*String* 部には補助記号の系列を持った文法である。*String* 部の補助記号の使い方によって i) *nested stack* 形 (ns), ii) *stack* 形 (s), iii) *pushdown* 形 (pd) *String Grammar* (SG) に分類する。

本報告では、SG が生成する言語の族のレベルについて述べる。まず、*Core* 部が  $\epsilon$ -free CF 形である、pd 形 SL は  $\epsilon$ -free CFL の族を真に含み、ns 形 SL は CSL の族と一

致することを明らかにする。つぎに、 $\varepsilon$ を許したCF形の生成規則であるならば、pd形SLは *recursively enumerable set* の族になることを示す。さらにSLの族はAF Lになることを述べる。

## §2. 諸定義

[定義1] ns形SGを  $G_n = (V, \Sigma, \Gamma, P, \sigma, z)$  とする。i)  $V$ ; 記号の有限集合, ii)  $\Sigma$ ; 終端記号の有限集合, iii)  $\Gamma$ ; 補助記号の有限集合, iv)  $P$ ;  $(V - \Sigma) \times \Gamma \times V^+ \times \Gamma^* \times D$  の部分集合で生成規則の集合を示す。 $(\xi, A, u, \alpha, i) \in P$  を  $(\xi, A) \rightarrow (u, \alpha, i)$  と記す。ここに、 $\xi \in V - \Sigma$ ,  $A \in \Gamma$ ,  $u \in V^+$ ,  $\alpha \in \Gamma^*$ ,  $i \in D = \{-1, 0, 1\}$  である。さらに、 $(\xi, A) \rightarrow (u, \varepsilon, i)$  のとき、 $u \notin V - \Sigma$  であり、 $A$  は String 部の最左端の補助記号である。v)  $\sigma \in V - \Sigma$ ; core部の初期記号, vi)  $z \in \Gamma$ ; String部の初期記号である。

[定義2] S形SGを  $G_s = (V, \Sigma, \Gamma, P, \sigma, z)$  とする。 $G_s$  は生成規則がつぎのように制限されたns形SGである。すなわち、 $P = P_1 \cup P_2$ ,  $P_1 \subseteq (V - \Sigma) \times \Gamma \times V^+ \times \Gamma^*$ ,  $P_2 \subseteq (V - \Sigma) \times \Gamma \times V^+ \times D$ ,  $D = \{-1, 0, 1\}$ 。ただし、 $P_1$ の生成規則は最左端にある補助記号に適用されるのみである。

[定義3] pd形SGを  $G_p = (V, \Sigma, \Gamma, P, \sigma, z)$  と

する。  $G_p$  は  $P$  として定義 2 の  $P_i$  の形のもののみをもつ  $S$  形  $S$   $G$  である。

[定義 4]  $SG, G=(V, \Sigma, P, P, \alpha, \Sigma)$  において,  $x, y \in V^*, \xi \in V - \Sigma, A \in P, \alpha, \beta, \gamma, \delta \in P^*, u \in V^+$  なるとき, もし,  $(\xi, A) \rightarrow (u, \beta, i)$  が  $P$  に含まれていれば  $(x \xi y, \alpha \uparrow A \gamma) \Rightarrow (x w y, \delta)$  なる関係  $\Rightarrow$  が成立つ。ただし,  $i = -1$  ならば,  $\delta = A_0 \cdots A_{j-1} \uparrow A_j \beta \gamma$ ,  $i = 0$  ならば,  $\delta = \alpha \uparrow \beta \gamma$ ,  $i = +1$  ならば,  $\delta = \alpha B_0 \uparrow B_1 \cdots B_j \gamma$ , さらに  $\beta = \varepsilon$ ,  $i = +1$  ならば,  $\delta = \alpha C_0 \uparrow C_1 \cdots C_j$ 。ここに,  $\alpha = A_0 \cdots A_j$ ,  $\beta = B_0 \cdots B_j$ ,  $\gamma = C_0 \cdots C_j$ , また,  $\Rightarrow$  の *reflective, transitive closure* を  $\xRightarrow{*}$  とする。

[定義 5]  $SG, G=(V, \Sigma, P, P, \alpha, \Sigma)$  によって生成される言語を  $L(G)$  とすれば,  $L(G) = \{w \mid (\alpha, \uparrow \Sigma) \xRightarrow{*} (w, \uparrow \Sigma), w \in \Sigma^*\}$ 。  $n$   $S$  形  $SG, G_n$  によって生成された言語  $L(G_n)$  を  $n$   $S$  形  $SL$  (String language),  $G_s$  によって生成された言語  $L(G_s)$  を  $S$  形  $SL$ ,  $pd$  形  $SG, G_p$  によって生成された言語  $L(G_p)$  を  $pd$  形  $SL$  という。また,  $L(G_n), L(G_s), L(G_p)$  の族をそれぞれ,  $\mathcal{L}_n, \mathcal{L}_s, \mathcal{L}_p$  と記す。定義 1, 2, 3 より  $\mathcal{L}_p \subseteq \mathcal{L}_s \subseteq \mathcal{L}_n$  が成立つ。

[定義 6]  $SG, G=(V, \Sigma, P, P, \alpha, \Sigma)$  において, 生成規則のすべての元が下記の, i), ii) の条件を満たしているとき,

normal SG という。  $(\xi, A) \rightarrow (u, \alpha, i)$  において,

i)  $1 \leq |u| \leq 2$ , ii)  $0 \leq |\alpha| \leq 2$ . とくに区別するときば  $G_0 = (V_0, \Sigma_0, \Gamma_0, P_0, \sigma_0, Z_0)$  と記す。ここに,  $|w|$  は語  $w$  の長さを表わす。

[定理 1] 任意の SG,  $G = (V, \Sigma, \Gamma, P, \sigma, Z)$  が生成する言語を  $L(G)$  とする。このとき,  $L(G) = L(G_0)$  である normal SG,  $G_0$  が存在する。(証明略)

### § 3. ns 形 SG の性質

[例 1] ns 形 SG,  $G_n = (V, \Sigma, \Gamma, P, \sigma, Z)$  において  
 $P = \{(\sigma, Z) \rightarrow (\xi, Z'AZ, 1), (\xi, A) \rightarrow (\xi, AA', 1)$   
 $(\xi, A') \rightarrow (\xi, A, 1), (\xi, Z) \rightarrow (\eta, Z, -1)$   
 $(\eta, A) \rightarrow (\eta, A, -1), (\eta, Z') \rightarrow (\xi, Z', 1)$   
 $(\eta, Z') \rightarrow (b\xi, \varepsilon, 0), (\xi, A) \rightarrow (a\xi, \varepsilon, 1)$   
 $(\xi, A) \rightarrow (a, \varepsilon, 0)\}$

とすると,  $L(G_n) = \{ba^2^n \mid n \geq 1\}$

[補題 1] ns 形 SG,  $G_n = (V, \Sigma, \Gamma, P, \sigma, Z)$  が生成する言語  $L(G_n)$  を認識可能な nondeterministic lba  $A$  が存在する。

(証明) 任意の ns 形 SG,  $G_n$  が生成する言語  $L(G_n)$  の任意の元  $w$  に対して,  $w$  を生成する  $G_n$  の導出  $D$  が必ず  $\cup$  とつは存在している。そこで,  $D; (\sigma, \uparrow Z) = (w_0, \alpha \uparrow \beta, \varepsilon) \Rightarrow$

$(w_1, \alpha_1 \uparrow \beta_1 z) \Rightarrow \dots \Rightarrow (w_r, \alpha_r \uparrow \beta_r z) = (w, \uparrow z)$  とすれば, Core 部は  $\varepsilon$ -free の生成規則によって導出されていることより,  $1 = |w_0| \leq |w_1| \leq |w_2| \leq \dots \leq |w_r| = |w| = n$

また,  $L(G)$  の定義より,  $w$  は  $(\circ, \uparrow z) \xrightarrow{*} (w, \uparrow z)$  によって導出されるので, 導出  $D$  が完了するまでには, string 部の  $z$  の左側に書かれた補助記号系列はすべて消去されなければならない。しかも, ns 形  $SG$  の定義から  $P$  の元,  $(\xi, A) \rightarrow (\eta, \varepsilon, i)$ ,  $\xi, \eta \in V - \Sigma$ ,  $A \in P$ ,  $i \in D$  なる形の生成規則は許されていない。したがって, string 部を消去するためには  $(\xi, A) \rightarrow (u, \varepsilon, i)$ ,  $u \in V^+$  かつ  $u \notin V - \Sigma$  なる形の生成規則を使用しなければならない。そこで, もし string 部の長さが  $m (> n+1)$  のものが存在していたとすると,  $(\xi, A) \rightarrow (u, \varepsilon, i)$  の形の生成規則を少なくとも  $m$  回使用しなければならない。ところが  $m$  回使用すれば, Core 部の性質により  $|w| \geq m$  でなければならない。これは,  $|w| = n$  としたことに矛盾する。したがって,  $\max_{0 \leq i \leq r} |\alpha_i \beta_i| \leq n$  でなければならない。以上のことより, 長さ  $n$  の語  $w$  を生成するとき使用する string 部の長さは高々  $(n+1)$  であることから,  $L(G_n)$  を認識可能な nondeterministic lba  $A$  を構成することができる。

(図 7 参照)。nondeterministic lba  $A$  は入力 word  $w$  ( $|w| = n$ ) に対して, working tape は  $2n+1$  の長さで認識できる。

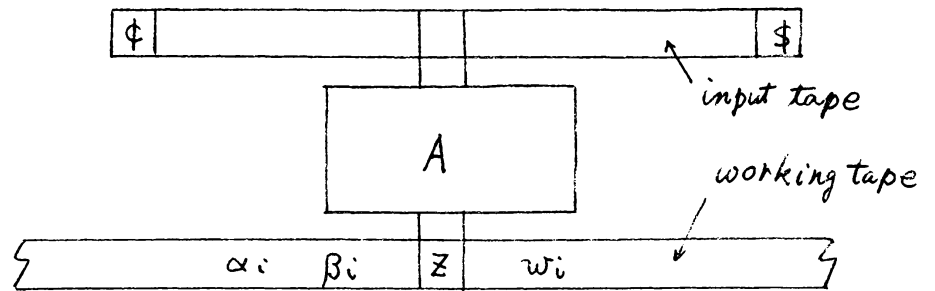


図 1. lba A.

[補題 2] ns 形  $SG, G_n$  は任意の  $CSG$ ,  $G$  が生成する言語  $L(G)$  を生成することができる。

(証明) 任意の  $CSG, G = (V, \Sigma, P, \sigma)$  は, 文献 1 よりつぎの 3 つの形の生成規則のみからなっていると考えてもよい。すなわち,  $\xi, \eta, \gamma, \zeta \in V - \Sigma, a \in \Sigma$  に対して,

$$i) \xi\eta \rightarrow \gamma\zeta, \quad ii) \xi \rightarrow \gamma\zeta, \quad iii) \xi \rightarrow a$$

これら, i), ii), iii) の形の生成規則を ns 形  $SG$  によってまねることを考える。 $CSG, G$  においては, i), ii) の形のみを使用して導出を行い, いったん, iii) の形の生成規則も適用したあとは, i), ii) の形の生成規則は使用しないで, 任意の word  $w$  を導出すると仮定しても一般性を失わない。ゆえに

$$\sigma = w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_l \Rightarrow \dots \Rightarrow w_n = w$$

の導出において,  $\sigma \xrightarrow{*} w_l$  までは, i), ii) の形の生成規則が適用されて,  $w_l \xrightarrow{*} w_n$  は iii) の形の生成規則のみが適用されて生成されたとする。そこで i), ii) の形の生成規則をつぎ

のようにして,  $ns$  形  $SG$ ,  $G_n$  でまねる.

$$i) (\sigma_s, A_\xi) \longrightarrow (\eta, A_r, 1)$$

$$(\eta, A_r) \longrightarrow (\sigma_s, A_\xi, 0), (\sigma_s \neq \eta)$$

$$ii) (\sigma_s, A_\xi) \longrightarrow (\sigma_s, A_r A_\xi, 0)$$

さらに,  $P$  の任意の元  $A$  に対して

$$(\sigma_s, A) \longrightarrow (\sigma_s, A, i), i \in D$$

また, 初期ステップとして,

$$(\sigma_s, Z) \longrightarrow (\sigma_s, A_0 Z, 0)$$

上のように構成すれば,  $\sigma \xrightarrow{*} w_l$  の導出を  $String$  部でまねることが出来る. すなわち,

$$(\sigma_s, \uparrow Z) \Rightarrow (\sigma_s, \uparrow A_0 Z) \xrightarrow{*} (\sigma_s, \overline{w_l} Z) \xrightarrow{*} (\sigma_s, w_l Z)$$

ただし,  $\overline{w_l} = A_\xi$  を表わす.

(注) もし,  $w_{i-1} \Rightarrow w_i$  は  $CSG$ ,  $G$  において,  $D$  の形の生成規則が適用されているとすると,

$$\begin{aligned} (\sigma_s, \overline{w_i} Z) &= (\sigma_s, \alpha \uparrow A_\xi A_n \beta Z) \Rightarrow (\eta, \alpha A_r \uparrow A_n \beta Z) \\ &\Rightarrow (\sigma_s, \alpha A_r \uparrow A_\xi \beta Z) = (\sigma_s, \overline{w_i} Z) \end{aligned}$$

つぎに, iii) の形の生成規則を  $ns$  形  $SG$ ,  $G$  でまねる.

$$iii) (\sigma_s, A_\xi) \longrightarrow (a\sigma'_s, \varepsilon, 0)$$

$$(\sigma'_s, A_\xi) \longrightarrow (a\sigma'_s, \varepsilon, 0)$$

$$(\sigma'_s, A_\xi) \longrightarrow (a, \varepsilon, 0)$$

$CSG$  の iii) の形の生成規則は  $CF$  形だから,  $w_l \xrightarrow{*} w_n$  は



iii) の形の生成規則を使用し、最左端導出によって生成されたと仮定してもよい。上のように構成すれば、String 部の左端から順に、CSG で、 $(\xi \rightarrow a)$  によって  $\xi$  が  $a$  に書換えられるのに対して、SG では、Core 部  $\lambda a$  を生成してゆく。このとき、String 部の記号は左から順に消去され、最後に  $\bar{z}$  だけになり、Core 部には  $w_n$  が生成される。よって補題 2 が成立つ。

[定理 2] ns 形 SG と CSG とは等価である。

(証明) 補題 1, 2 よりいえる。

### § 5. pd 形 SG の性質

[定義 7] pd 形 SG,  $G_p = (V, \Sigma, \Gamma, P, \sigma, \bar{z})$  が最左端導出によって生成する言語を  $L_1(G_p)$  とする。また、 $L_1(G_p) = L(G_p)$  となる  $L(G_p)$  の族を  $\mathcal{L}_p'$  とする。

例 2] pd 形 SG,  $G_p = (V, \Sigma, \Gamma, P, \sigma, \bar{z})$  において、

$$P = \{(\sigma, \bar{z}) \rightarrow (\xi \eta, A\bar{z}, 0), (\xi, A) \rightarrow (ab, \varepsilon, 0)$$

$$(\xi, A) \rightarrow (a\xi b, AB, 0), (\eta, B) \rightarrow (c\eta, \varepsilon, 0)$$

$$(\eta, B) \rightarrow (c, \varepsilon, 0)\}$$

とすると、 $L(G_p) = L_1(G_p) = \{a^n b^n c^n \mid n \geq 1\}$  となる。

[定理 3]  $\mathcal{L}_p'$  は  $\varepsilon$ -free CFL の族を真に含む。

(証明) 任意の  $\varepsilon$ -free CFG  $G' = (V', \Sigma', P', \sigma')$  とする。

$(\xi \rightarrow w) \in P'$  のときかつそのときにかぎり  $(\xi, \bar{z}) \rightarrow (w, \bar{z},$

$0) \in P$  とすれば,  $L(G_p) = L(G')$  は  $pd$  形  $SG$ ,  $G = (V, \Sigma, P, \sigma, \tau)$  を構成できる。しかも,  $P$  の core 部が  $\varepsilon$ -free CF 形だから,  $L(G_p)$  のすべての元は最左端導出によって生成されるように構成できる。したがって,  $L(G_p) = L_1(G_p)$ 。つぎに例2により  $\{a^n b^n c^n \mid n \geq 1\}$  は CFL でない CSL であり,  $\mathcal{L}_p'$  に含まれるから定理3がいえる。

[補題3]  $pd$  形  $SG$  において, Core 部に  $\varepsilon$  を許したものを,  $G^*$  とする。すなわち,  $G^* = (V, \Sigma, P, P^*, \sigma, \tau)$  の生成規則の集合  $P^*$  は  $(V - \Sigma) \times P \times V^* \times P^*$  の部分集合である。このとき, 任意の CFG, CFL, C に対して文献2が定義する  $L_c(G)$  と  $L_1(G_p^*)$  とが等しい,  $pd$  形  $SG$ ,  $G_p^*$  が存在する。

(証明) 文献2と同じ記号を用いる。  $L_c(G) = g_G(\{0\}, C \cap P^*) \cap \Sigma^*$  である。  $L_c(G)$  の任意の元  $w$  とすると,  $w = g_G(\sigma, c)$ ,  $c \in C$  が存在する。すなわち,

$$\sigma = w_0 \xrightarrow{p_1} w_1 \xrightarrow{p_2} \cdots \xrightarrow{p_r} w_r = w, \quad c = p_1 p_2 \cdots p_r$$

なる導出が存在している。ここに,  $w_{i-1} \xrightarrow{p_i} w_i$  は,  $w_{i-1}$  から  $w_i$  の導出は,  $p_i$  の生成規則を  $w_{i-1}$  の最左端の変数に適用して生成されていることを示す。一方,  $C = L(G')$  は CFG,  $G'$  が存在する。それと,  $G' = (V', \Sigma', P', \sigma')$ ,  $P' = \{\xi' \rightarrow w'\}$  とする。そこで,  $pd$  形  $SG$ ,  $G_p^* = (V, \Sigma, P^*, P, \sigma, \tau)$  と  $P = V'$ ,  $P = \{(\sigma, \tau) \rightarrow (\sigma, \sigma\tau, 0)\} \cup \{(\xi, \xi') \rightarrow (\xi, u', 0) \mid \xi$

$\in V - \Sigma, \xi' \rightarrow v' \in P' \} \cup \{(\xi, p) \rightarrow (v, \varepsilon, 0) \mid p: \xi \rightarrow v \in P\}$  のように構成すれば,  $L_c(G)$  における  $w$  に対する導出に対応して,  $G_S$  において  $w$  の導出が存在する。

$$(\alpha, z) \xRightarrow{*} (\alpha, \alpha' z) = (w_0, w'_0) \xRightarrow{*} (w_r, w'_r) = (w, z)$$

また, この逆も成立つ. したがって,  $L_c(G) = L_1(G_p^*)$  なる  $pd$  形  $SG$ ,  $G_p^*$  が存在する.

[補題 4] 任意の recursively enumerable set  $U$  に対して,  $U = L_c(G)$  を満す,  $CFG, G, CFL, C$  とが存在する.

(証明) 文献 1 の定理 2.1 による.

[定理 5] 任意の recursively enumerable set  $U$  に対して,  $U = L_1(G_p^*)$  を満す  $pd$  形  $SG$ ,  $G_p^*$  が存在する.

(証明) 補題 3, 4 による.

## § 6. $SL$ の Closure property

[補題 6]  $L_p, L_s, L_n$  は  $\cup, \cdot, +$  の演算のもとでとれている.

[補題 7]  $L_p, L_s, L_n$  は regular set との intersection のもとでとれている.

[補題 8]  $L_p, L_s, L_n$  は  $\varepsilon$ -free homomorphism, inverse homomorphism のもとでとれている.

[定理 5]  $L_p, L_s, L_n$  は AFL である.

(証明) 補題 5, 6, 7 による

## § 7. あとがき

今後の課題として, i)  $\mathcal{L}_p, \mathcal{L}_s, \mathcal{L}_n$  との間に真の包含関係が成立つか. ii) 生成規則の適用順序が制御されている種々の既存の文法との対応関係について調らべる必要がある。

謝辞, 日頃熱心に御指導賜る研究室の皆様, 東北大学本多波雄教授に感謝する。

## § 8 文献

- 1) Kuroda, S.Y. "Classes of Languages and Linear-Bounded Automata" *Inf & Cont.* Vol 7, p207-223
- 2) Ginsburg, S. "Control sets on Grammars" *mathematical System Theory*, Vol. 2 p157-177 (1968)